

## Lecture 7

## Optimistic/predictive regret minimization via online optimization

Instructor: Gabriele Farina\*

## 1 Predictive regret minimization

So far, we have talked extensively about regret minimization algorithms, which we modeled as object with the following interface:

- `NEXTSTRATEGY()` returns a strategy  $\mathbf{x}^t \in \mathcal{X} \subseteq \mathbb{R}^n$ ;
- `OBSERVELOSS( $\ell^t$ )` receives a *utility vector*  $\ell^t$  that is meant to evaluate the strategy  $\mathbf{x}^t$  that was last output.

However, a recent trend in online learning has been concerned with constructing devices that can incorporate *predictions* of the next utility vector  $\ell^t$  in the decision making [Chiang et al., 2012, Rakhlin and Sridharan, 2013a,b]. We call these devices *predictive* regret minimizers.

We incorporate predictivity by modifying the interface above. In particular, for a predictive regret minimizer we modify `NEXTSTRATEGY` to now be as in the next definition.

**Definition 1.1** (Predictive regret minimizer). Let  $\mathcal{X}$  be a set. A *predictive regret minimizer* for  $\mathcal{X}$  is an algorithm that interacts with the environment through two operations:

- `NEXTSTRATEGY( $\mathbf{m}^t$ )` returns the next strategy  $\mathbf{x}^t \in \mathcal{X}$ , given a *prediction*  $\mathbf{m}^t \in \mathbb{R}^n$  of the next utility  $\ell^t$ ;
- `OBSERVELOSS( $\ell^t$ )` receives a *utility vector*  $\ell^t$  that is meant to evaluate the strategy  $\mathbf{x}^t$  that was last output. Specifically, the device incurs a utility equal to  $(\ell^t)^\top \mathbf{x}^t$ . As we mentioned, the utility vector  $\ell^t$  can depend on all past strategies that were output by the regret minimizer (even including  $\mathbf{x}^t$ , assuming the latter is deterministic).

Just like for regular (*i.e.*, non-predictive) regret minimizers, the quality metric for a predictive regret minimizer is its cumulative regret, defined as the quantity

$$R^T := \max_{\hat{\mathbf{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^T (\ell^t)^\top \hat{\mathbf{x}} - (\ell^t)^\top \mathbf{x}^t \right\}. \quad (1)$$

The goal for a “good” predictive regret minimizer should guarantee a superior regret bound than a non-predictive regret minimizer, especially if  $\mathbf{m}^t \approx \ell^t$  at all times  $t$ . Algorithms exist that can guarantee this. For instance, it is always possible to guarantee that

$$R^T = O\left(1 + \sum_{t=1}^T \|\ell^t - \mathbf{m}^t\|^2\right)$$

\*Computer Science Department, Carnegie Mellon University. ✉ gfarina@cs.cmu.edu.

when  $\mathcal{X}$  is convex and compact, which implies that the regret stays *constant* when  $\mathbf{m}^t$  is clairvoyant. In fact, even stronger regret bounds can be attained, as we discuss next.

## 1.1 Regret bounded by Variation in Utilities (RVU)

A popular definition for a desirable regret goal for a predictive regret minimizer was given by Syrgkanis et al. [2015]. We restate it here with minor modifications.

**Definition 1.2** (RVU regret bound). A predictive regret minimizers satisfies the *RVU regret bound* with parameters  $\alpha, \beta, \gamma \in \mathbb{R}_{\geq 0}$  with respect to norm  $\|\cdot\|$  if at all times  $T$ ,

$$R^T \leq \alpha + \beta \sum_{t=1}^T \|\ell^t - \mathbf{m}^t\|_*^2 - \gamma \sum_{t=2}^T \|\mathbf{x}^t - \mathbf{x}^{t-1}\|^2. \quad (2)$$

In Section 1.2 we show that predictive regret minimizers that satisfy the RVU regret bound can be used in self play to converge to a bilinear saddle point at the accelerated convergence (saddle point gap) rate of  $O(1/T)$ . We will then introduce predictive regret minimizers for generic convex and compact sets in Section 2 and show that they satisfy the RVU regret bound.

## 1.2 Accelerated convergence to saddle points via predictive regret minimization

We have seen in Lecture 3 that the idea behind using regret minimization to converge to a bilinear saddle point

$$\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \mathbf{x}^\top \mathbf{A} \mathbf{y} \quad (3)$$

is to use *self play*. Back then, we instantiated two regret minimization algorithms,  $\mathcal{R}_X$  and  $\mathcal{R}_Y$ , for the domains of the maximization and minimization problem, respectively. At each time  $t$  the two regret minimizers output strategies  $\mathbf{x}^t$  and  $\mathbf{y}^t$ , respectively. Then, they receive as feedback the vectors  $\ell_x^t, \ell_y^t$ , defined as

$$\ell_x^t := \mathbf{A} \mathbf{y}^t, \quad \ell_y^t := -\mathbf{A}^\top \mathbf{x}^t, \quad (4)$$

where  $\mathbf{A}$  is Player 1's payoff matrix (see also Figure 1).

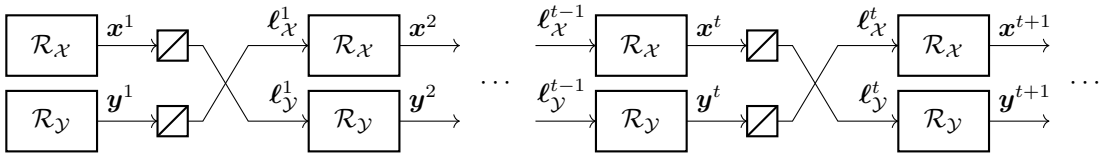


Figure 1: The flow of strategies and utilities in regret minimization for games. The symbol  $\square$  denotes computation/construction of the utility vector.

We then proved that the average strategies

$$\bar{\mathbf{x}}^T := \frac{1}{T} \sum_{t=1}^T \mathbf{x}^t, \quad \bar{\mathbf{y}}^T := \frac{1}{T} \sum_{t=1}^T \mathbf{y}^t$$

converge to a saddle point of (3) in the sense that the saddle-point gap of  $(\bar{\mathbf{x}}^T, \bar{\mathbf{y}}^T)$  vanishes according to

$$\gamma(\bar{\mathbf{x}}^T, \bar{\mathbf{y}}^T) \leq \frac{R_X^T + R_Y^T}{T}.$$

As most non-predictive regret minimizers guarantee  $O(\sqrt{T})$  regret in the worst case, the self-play scheme defined above guarantees convergence to Nash equilibrium at the rate  $O(1/\sqrt{T})$ .

We now show that by using *predictive* regret minimizers that satisfy the RVU bound (2), we can accelerate convergence from  $O(1/\sqrt{T})$  to  $O(1/T)$ . The key is to use as predictions the previous utility vectors, that is,

$$\mathbf{m}_{\mathcal{X}}^t := \boldsymbol{\ell}_{\mathcal{X}}^{t-1}, \quad \mathbf{m}_{\mathcal{Y}}^t := \boldsymbol{\ell}_{\mathcal{Y}}^{t-1}.$$

Then, the gap  $\gamma(\bar{\mathbf{x}}^T, \bar{\mathbf{y}}^T)$  satisfies

$$\begin{aligned} \gamma(\bar{\mathbf{x}}^T, \bar{\mathbf{y}}^T) &\leq \frac{R_{\mathcal{X}}^T + R_{\mathcal{Y}}^T}{T} \\ &\leq \frac{1}{T} \left( \alpha + \beta \sum_{t=1}^T \|\mathbf{A}\mathbf{y}^t - \mathbf{A}\mathbf{y}^{t-1}\|_*^2 - \gamma \sum_{t=2}^T \|\mathbf{x}^t - \mathbf{x}^{t-1}\|^2 \right) \\ &\quad + \frac{1}{T} \left( \alpha + \beta \sum_{t=1}^T \|\mathbf{A}^\top \mathbf{x}^t - \mathbf{A}^\top \mathbf{x}^{t-1}\|_*^2 - \gamma \sum_{t=2}^T \|\mathbf{y}^t - \mathbf{y}^{t-1}\|^2 \right) \\ &\leq \frac{2\alpha}{T} + \frac{\beta \|\mathbf{A}\|_{\text{op}}^2 - \gamma}{T} \sum_{t=1}^T \|\mathbf{y}^t - \mathbf{y}^{t-1}\|^2 + \frac{\beta \|\mathbf{A}\|_{\text{op}}^2 - \gamma}{T} \sum_{t=1}^T \|\mathbf{x}^t - \mathbf{x}^{t-1}\|^2, \end{aligned} \quad (5)$$

where in the second inequality we plugged in the RVU regret bounds for  $\mathcal{R}_{\mathcal{X}}$  and  $\mathcal{R}_{\mathcal{Y}}$  (Definition 1.2) and the third inequality follows by noting that the operator norm  $\|\cdot\|_{\text{op}}$  of a linear function is equal to the operator norm of its transpose.

Inequality (5) immediately implies that if  $\|\mathbf{A}\|_{\text{op}}^2 \leq \gamma/\beta$  then the saddle point gap  $\gamma(\bar{\mathbf{x}}^T, \bar{\mathbf{y}}^T)$  vanishes at the rate  $O(1/T)$ . Since rescaling  $\mathbf{A}$  in (3) does not change the set of saddle points, we can always rescale  $\mathbf{A}$  before applying the self-play algorithm above to guarantee accelerated  $O(1/T)$  convergence.

## 2 Predictive FTRL and predictive OMD

*Follow-the-regularized-leader (FTRL)* [Shalev-Shwartz and Singer, 2007] and *online mirror descent (OMD)* are the two best known oracles for the online linear optimization problem. Their *predictive* variants are relatively new and can be traced back to the works by Rakhlin and Sridharan [2013a] and Syrgkanis et al. [2015].

Algorithms 1 and 2 give pseudocode for predictive FTRL and predictive OMD. The non-predictive variants of FTRL and OMD algorithms correspond to predictive FTRL and predictive OMD when the prediction  $\mathbf{m}^t$  is set to the  $\mathbf{0}$  vector at all  $t$ . In both algorithm,  $\eta > 0$  is an arbitrary step size parameter,  $\mathcal{X} \subseteq \mathbb{R}^n$  is a convex and compact set, and  $\varphi : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$  is a differentiable and 1-strongly convex regularizer (with respect to some norm  $\|\cdot\|$ ). The symbol  $D_\varphi(\cdot \|\cdot)$  used in OMD denotes the *Bregman divergence* associated with  $\varphi$ , a standard surrogate notion of distance in convex optimization, defined as

$$D_\varphi(\mathbf{x} \|\mathbf{c}) := \varphi(\mathbf{x}) - \varphi(\mathbf{c}) - \nabla \varphi(\mathbf{c})^\top (\mathbf{x} - \mathbf{c}) \quad \forall \mathbf{x}, \mathbf{c} \in \mathcal{X}.$$

**Remark 2.1.** The divergence  $D_\varphi(\mathbf{x} \|\mathbf{c})$  is a generalization of the typical notion of distance between  $\mathbf{x}$  and  $\mathbf{c}$ . When the regularizer  $\varphi$  is set to  $\varphi(\cdot) = \frac{1}{2} \|\cdot\|_2^2$  (which is 1-strongly convex with respect to the

---

**Algorithm 1:** (Predictive) FTRL

---

**Data:**  $\mathcal{X} \subseteq \mathbb{R}^n$  convex and compact set  
 $\varphi : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$  strongly convex regularizer  
 $\eta > 0$  step-size parameter

- 1  $\mathbf{L}^0 \leftarrow \mathbf{0} \in \mathbb{R}^n$

---

  - 2 **function** NEXTSTRATEGY( $\mathbf{m}^t$ )  
    [▷ Set  $\mathbf{m}^t = \mathbf{0}$  for non-predictive version]
  - 3 **return**  $\arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} \left\{ (\mathbf{L}^{t-1} + \mathbf{m}^t)^\top \hat{\mathbf{x}} - \frac{1}{\eta} \varphi(\hat{\mathbf{x}}) \right\}$

---

  - 4 **function** OBSERVEUTILITY( $\ell^t$ )
  - 5  $\mathbf{L}^t \leftarrow \mathbf{L}^{t-1} + \ell^t$
- 

---

**Algorithm 2:** (Predictive) OMD

---

**Data:**  $\mathcal{X} \subseteq \mathbb{R}^n$  convex and compact set  
 $\varphi : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$  strongly convex regularizer  
 $\eta > 0$  step-size parameter

- 1  $\mathbf{z}^0 \in \mathcal{X}$  such that  $\nabla \varphi(\mathbf{z}^0) = \mathbf{0}$

---

  - 2 **function** NEXTSTRATEGY( $\mathbf{m}^t$ )  
    [▷ Set  $\mathbf{m}^t = \mathbf{0}$  for non-predictive version]
  - 3 **return**  $\arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} \left\{ (\mathbf{m}^t)^\top \hat{\mathbf{x}} - \frac{1}{\eta} D_\varphi(\hat{\mathbf{x}} \parallel \mathbf{z}^{t-1}) \right\}$

---

  - 4 **function** OBSERVEUTILITY( $\ell^t$ )
  - 5  $\mathbf{z}^t \leftarrow \arg \max_{\hat{\mathbf{z}} \in \mathcal{X}} \left\{ (\ell^t)^\top \hat{\mathbf{z}} - \frac{1}{\eta} D_\varphi(\hat{\mathbf{z}} \parallel \mathbf{z}^{t-1}) \right\}$
- 

Euclidean norm), then

$$\begin{aligned} D_\varphi(\mathbf{x} \parallel \mathbf{c}) &= \varphi(\mathbf{x}) - \varphi(\mathbf{c}) - \nabla \varphi(\mathbf{c})^\top (\mathbf{x} - \mathbf{c}) \\ &= \frac{1}{2} \|\mathbf{x}\|_2^2 - \frac{1}{2} \|\mathbf{c}\|_2^2 - \mathbf{c}^\top (\mathbf{x} - \mathbf{c}) \\ &= \frac{1}{2} \|\mathbf{x}\|_2^2 + \frac{1}{2} \|\mathbf{c}\|_2^2 - \mathbf{c}^\top \mathbf{x} \\ &= \frac{1}{2} \|\mathbf{x} - \mathbf{c}\|_2^2 \end{aligned}$$

and so we recover that  $D_\varphi(\mathbf{x} \parallel \mathbf{c})$  is half of the squared Euclidean distance between  $\mathbf{x}$  and  $\mathbf{c}$ .

**Remark 2.2.** In light of the above remark, when OMD is set up with  $\varphi(\cdot) = \|\cdot\|_2^2$ , the computation of  $\mathbf{z}^{t+1}$  (Line 5 of Algorithm 2) amounts to a step of projected gradient ascent. Indeed,

$$\begin{aligned} \arg \max_{\hat{\mathbf{z}} \in \mathcal{X}} \left\{ (\ell^t)^\top \hat{\mathbf{z}} - \frac{1}{2\eta} D_\varphi(\hat{\mathbf{z}} \parallel \mathbf{z}^{t-1}) \right\} &= \arg \max_{\hat{\mathbf{z}} \in \mathcal{X}} \left\{ (\ell^t)^\top \hat{\mathbf{z}} - \frac{1}{2\eta} \|\hat{\mathbf{z}} - \mathbf{z}^{t-1}\|_2^2 \right\} \\ &= \arg \max_{\hat{\mathbf{z}} \in \mathcal{X}} \left\{ (\eta \ell^t)^\top \hat{\mathbf{z}} - \frac{1}{2} \|\hat{\mathbf{z}} - \mathbf{z}^{t-1}\|_2^2 \right\} \\ &= \arg \max_{\hat{\mathbf{z}} \in \mathcal{X}} \left\{ (\mathbf{z}^{t-1} + \eta \ell^t)^\top \hat{\mathbf{z}} - \frac{1}{2} \|\hat{\mathbf{z}}\|_2^2 \right\} \\ &= \arg \max_{\hat{\mathbf{z}} \in \mathcal{X}} \left\{ \frac{1}{2} \|\hat{\mathbf{z}} - (\mathbf{z}^{t-1} + \eta \ell^t)\|_2^2 \right\} \\ &= \text{Proj}_{\mathcal{X}}(\mathbf{z}^{t-1} + \eta \ell^t). \end{aligned}$$

**Remark 2.3.** In the non-predictive version of OMD,  $\mathbf{m}^t = \mathbf{0}$  at all times  $t$ . In that case, the proximal

step on Line 3 in Algorithm 2 reduces to

$$\mathbf{x}^t = \arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} \left\{ -\frac{1}{\eta} D_\varphi(\hat{\mathbf{x}} \parallel \mathbf{z}^{t-1}) \right\} = \arg \min_{\hat{\mathbf{x}} \in \mathcal{X}} D_\varphi(\hat{\mathbf{x}} \parallel \mathbf{z}^{t-1}) = \mathbf{z}^{t-1}.$$

## 2.1 Regret guarantee

Predictive FTRL and predictive OMD satisfy the following regret bound.

**Proposition 2.1.** Let  $\Omega$  denote the range of  $\varphi$  over  $\mathcal{X}$ , that is,  $\Omega := \max_{\mathbf{x}, \mathbf{x}' \in \mathcal{X}} \{\varphi(\mathbf{x}) - \varphi(\mathbf{x}')\}$ . At all times  $T$ , the regret cumulated by predictive FTRL (Algorithm 1) and predictive OMD (Algorithm 2) compared to *any* strategy  $\hat{\mathbf{x}} \in \mathcal{X}$  is bounded as

$$R^T \leq \frac{\Omega}{\eta} + \eta \sum_{t=1}^T \|\ell^t - \mathbf{m}^t\|_*^2 - \frac{1}{c\eta} \sum_{t=2}^T \|\mathbf{x}^t - \mathbf{x}^{t-1}\|^2,$$

where  $c = 4$  for FTRL and  $c = 8$  for OMD, and where  $\|\cdot\|_*$  denotes the dual of the norm  $\|\cdot\|$  with respect to which  $\varphi$  is 1-strongly convex. In other words, predictive FTRL satisfies the RVU regret bound with parameters  $(\Omega/\eta, \eta, 1/4\eta)$  and predictive OMD satisfies the RVU regret bound with parameters  $(\Omega/\eta, \eta, 1/8\eta)$ .

**Remark 2.4.** Proposition 2.1 immediately implies that, by appropriately setting the step size parameter (for example,  $\eta = T^{-1/2}$ ), predictive FTRL and predictive OMD guarantee  $R^T = O(T^{1/2})$  at all times  $T$ . Hence, predictive FTRL and predictive OMD are regret minimizers.

## 3 Distance-generating functions for tree-form decision problems

Predictive FTRL and predictive OMD, as described above, can be applied to any convex and compact set  $\mathcal{X}$ , provided a 1-strongly convex regularizer  $\varphi$  for  $\mathcal{X}$  has been chosen. A safe choice is always  $\varphi(\cdot) = \frac{1}{2} \|\cdot\|_2^2$ , which is 1-strongly convex with respect to the Euclidean norm  $\|\cdot\|_2$ . In that case, we have seen earlier that (non-predictive) OMD reduces to online gradient ascent. However, the choice of regularizer  $\varphi$  has practical implications, because it can make solving the different optimization subproblems involved in the algorithms (for example, Line 5 in Algorithm 2) easy or hard in practice depending on the choice.

Ideally, the choice of regularizer  $\varphi$  makes the computation of the following two quantities as easy as possible. In particular we define the following.

**Definition 3.1** (“Nice” regularizer). Let  $\mathcal{X} \subseteq \mathbb{R}^n$  be convex and compact. We say that a 1-strongly convex regularizer  $\varphi : \mathcal{X} \rightarrow \mathbb{R}$  is “nice” if the following quantities can both be computed in linear time in the dimension  $n$ :

- the *gradient*  $\nabla\varphi(\mathbf{x})$  of  $d$  at any point  $\mathbf{x} \in \mathcal{X}$ ; and

- the gradient of the convex conjugate  $\varphi^*$  of  $\varphi$  at any point  $\mathbf{g} \in \mathbb{R}^n$ :

$$\nabla\varphi^*(\mathbf{g}) = \arg \max_{\mathbf{x} \in \mathcal{X}} \{\mathbf{g}^\top \mathbf{x} - \varphi(\mathbf{x})\}.$$

When the regularizer is nice, then the optimization subproblem on line Line 3 of predictive FTRL can be computed in linear time by noticing that

$$\arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} \{(\eta \mathbf{L}^{t-1} + \eta \mathbf{m}^t)^\top \hat{\mathbf{x}} - \varphi(\hat{\mathbf{x}})\} = \nabla\varphi^*(\eta \mathbf{L}^{t-1} + \eta \mathbf{m}^t).$$

For Line 3 in predictive OMD, we have

$$\begin{aligned} \arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} \left\{ (\mathbf{m}^t)^\top \hat{\mathbf{x}} - \frac{1}{\eta} D_\varphi(\hat{\mathbf{x}} \| \mathbf{z}^{t-1}) \right\} &= \arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} \{(\eta \mathbf{m}^t)^\top \hat{\mathbf{x}} - D_\varphi(\hat{\mathbf{x}} \| \mathbf{z}^{t-1})\} \\ &= \arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} \{(\eta \mathbf{m}^t)^\top \hat{\mathbf{x}} - \varphi(\hat{\mathbf{x}}) + \nabla\varphi(\mathbf{z}^{t-1})^\top \hat{\mathbf{x}}\} \\ &= \arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} \{(\eta \mathbf{m}^t + \nabla\varphi(\mathbf{z}^{t-1}))^\top \hat{\mathbf{x}} - \varphi(\hat{\mathbf{x}})\} \\ &= \nabla\varphi^*(\eta \mathbf{m}^t + \nabla\varphi(\mathbf{z}^{t-1})), \end{aligned}$$

which can therefore be evaluated in linear time when  $\varphi$  is nice. Similarly, for Line 5 we have, using the same derivation,

$$\arg \max_{\hat{\mathbf{z}} \in \mathcal{X}} \left\{ (\boldsymbol{\ell}^t)^\top \hat{\mathbf{z}} - \frac{1}{\eta} D_\varphi(\hat{\mathbf{z}} \| \mathbf{z}^{t-1}) \right\} = \nabla\varphi^*(\eta \boldsymbol{\ell}^t + \nabla\varphi(\mathbf{z}^{t-1})).$$

For many sets of interest in game theory, “nice” regularizer are known. For example, for a simplex domain a very appealing regularizer is the negative entropy distance generating function

$$\Delta^n \ni (x_1, \dots, x_n) \mapsto \sum_{i=1}^n x_i \log x_i.$$

The construction of a “nice” regularizer for sequence-form strategy polytopes is significantly more involved, but a modification/generalization of negative entropy with specific weights can be shown to work [Farina et al., 2021].

**Definition 3.2** (Dilatable global entropy). Consider a tree-form sequential decision process, with the usual notation.<sup>a</sup> The *dilatable global entropy distance generating function*  $\tilde{\varphi}$  is the function  $\tilde{\varphi} : Q \rightarrow \mathbb{R}_{\geq 0}$  defined as

$$\tilde{\varphi} : Q \ni \mathbf{x} \mapsto \sum_{j \in \mathcal{J}} \sum_{a \in A_j} w_{ja} \mathbf{x}[ja] \log \mathbf{x}[ja],$$

where each coefficient  $\gamma_j \geq 1$  ( $j \in \mathcal{J}$ ) is defined resursively as

$$\gamma_j := 1 + \max_{a \in A_j} \left\{ \sum_{j' \in \mathcal{J}: p_{j'} = ja} \gamma_{j'} \right\} \quad \forall j \in \mathcal{J}, \quad (6)$$

and each  $w_{ja} \geq 1$  ( $ja \in \Sigma$ ) is defined recursively as

$$w_{ja} := \gamma_j - \sum_{j' \in \mathcal{J}: p_{j'} = ja} \gamma_{j'} \quad \forall ja \in \Sigma.$$

---

<sup>a</sup>In particular, remember that  $\mathcal{J}$  denotes the set of decision points, and given any  $j \in \mathcal{J}$ ,  $A_j$  is the set of actions available at  $j$ . The parent sequence of  $j$ , denoted  $p_j$ , is the last sequence (decision point-action pair) on the path from the root of the tree-form decision process to  $j$ .

**Theorem 3.1.** The dilatable global entropy function  $\tilde{\varphi} : Q \rightarrow \mathbb{R}_{\geq 0}$  is a regularizer for the sequence-form polytope  $Q$ . It is 1-strongly convex with respect to the  $\ell_2$  (Euclidean) norm, and  $(1/\|Q\|_1)$ -strongly convex with respect to the  $\ell_1$  norm, where  $\|Q\|_1$  is the  $\ell_1$ -diameter of  $Q$ , that is,  $\|Q\|_1 := \max_{\mathbf{q} \in Q} \|\mathbf{q}\|_1$ .

## References

- Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Proceedings of the Conference on Learning Theory (COLT)*, 2012.
- Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Proceedings of the Conference on Learning Theory (COLT)*, 2013a.
- Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2013b.
- Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems*, 2015.
- Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.
- Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Better regularization for sequential decision spaces: Fast convergence rates for Nash, correlated, and team equilibria. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2021.